Computers in Industry, Volume 82, October 2016, Pages 273–289.
DOI 10.1016/j.compind.2016.05.006

Design, Modelling, Simulation and Integration of Cyber Physical Systems: Methods and Applications

P. Hehenberger[a]*, B. Vogel-Heuser[b], D. Bradley[c], B. Eynard[d], T. Tomiyama[e], S. Achiche[f]

[a] Institute of Mechatronic Design and Production, Johannes Kepler University Linz, Austria

[b] Institute of Automation and Information Systems, Technische Universität München, Garching, Germany

[c] Abertay University, Dundee, UK

[d] Sorbonne Universités, Université de Technologie de Compiègne, Department of Mechanical Systems Engineering, France

[e] Cranfield University, Bedfordshire, UK

[f] École Polytechnique de Montréal, Québec, Canada

*Corresponding author: E-mail address: peter.hehenberger@jku.at

Published by Elsevier. This is the Author Accepted Manuscript issued with:
Creative Commons Attribution Non-Commercial No Derivatives License (CC:BY:NC:ND 4.0).
The final published version (version of record) is available online at 10.1016/j.compind.2016.05.006
Please refer to any applicable publisher terms of use.

Abstract:

The main drivers for the development and evolution of Cyber Physical Systems (CPS) are the reduction of development costs and time along with the enhancement of the designed products. The aim of this survey paper is to provide an overview of different types of system and the associated transition process from mechatronics to CPS and cloud-based (IoT) systems. It will further consider the requirement that methodologies for CPS-design should be part of a multi-disciplinary development process within which designers should focus not only on the separate physical and computational components, but also on their integration and interaction. Challenges related to CPS-design are therefore considered in the paper from the perspectives of the physical processes, computation and integration respectively. Illustrative case studies are selected from different system levels starting with the description of the overlaying concept of Cyber Physical Production Systems (CPPSs). The analysis and evaluation of the specific properties of a sub-system using a condition monitoring system, important for the maintenance purposes, is then given for a wind turbine.

Keywords:
Cyber Physical Systems, Mechatronics, Design, Modelling, CPS-paradigm, system classification

Highlights:

- Presentation of a systematic classification of systems and new CPS paradigms.
- Analyses of literature conducted across a range of different perspectives.
- A systematic review of CPS-Design literature was carried out, with an emphasis on the design, modelling, simulation and integration of CPS.
- An architectural and behavioural paradigm for CPS.
- Compilation of different viewpoints referring to applications at different levels of granularity.

1. Introduction

The main drivers for the development and evolution of Cyber Physical Systems (CPS) are the reduction of development costs and time along with the enhancement of the designed products. This involves the virtualization of the product to improve its design, to support verification and validation and enhance its production and operation. Overall, virtualization enables more flexibility at lower cost across the different stages of development. The interaction between the designed product and the production systems plays an important role in the development of the Industry 4.0 (also called Advanced Manufacturing, Smart Manufacturing and Cyber Physical Production Systems) concept. Future trends, methods and models for the systems design process also have to be considered in relation to their role as enablers of transformation of traditional system paradigms, such as mechatronics, embedded intelligence, and automation systems, into Cyber Physical Systems or the global integration associated with the Internet of Things (IoT).

The concept of automation, an evolved concept of mechanisation, began to develop in the mid-1950s as the increasing ability to apply developments in instrumentation and computer technology to the control of systems suggested a shift towards an environment in which computers would become increasing responsible for the operation of a range of processes [Mac56, Car15]. The initial concepts of automation were generally associated with manufacturing processes, and the ability to remove people from many of those processes, resulting in what was envisioned as a safer environment producing products with greater efficiency and consistency. As for instance the replacement of electronic systems by processors such as the PDP-8 and PDP-11series from the mid-1960s onwards.

Since the development of the original concepts of manufacturing automation, computers have increasingly been adopted throughout the whole of industry for tasks covering the entirety of the design, development and manufacturing processes, as well as for associated tasks such as financial control, marketing and logistics. Further, the role of industrial computing has extended beyond its original area of manufacturing into fields such as healthcare where the ability to collect and manage large volumes of data has been assessed as having a potential value of billions of dollars [KKC13]. However, in many cases the expected results perhaps have not as yet materialised.

In considering the growth of computers as applied to industrial environments, there are a number of factors and issues that need to be taken into account in relation to next generation systems. In particular:

1. A shift from product based to information or knowledge based economies integrated with system level developments such as Cyber Physical Systems and the Internet of Things. In such economies, the ability to access, transfer and share relevant knowledge on the basis of both context and need is likely to redefine the nature of

systems, particularly when associated with "*manufacture on demand*" technologies such as 3D printing (additive manufacturing).

2. A refocusing on, and perhaps a redefinition of, the role of the user in a wide range of industry based activities in which the task differentiation takes better account of the associative and collaborative nature of activities. For instance, there is accumulating evidence to suggest that placing humans, with their short attention spans, into an environment dominated and controlled by computers results in a skills loss and reduced decision making capacity as well as in the ability to innovate.
3. A redefinition of the design processes to take account of increasing levels of system complexity and the inability of individuals, or even groups of individuals, to properly comprehend the nature of the system, or its potential failure modes [Lev12].

Over recent years, significant effort has been put in to understanding the relationship of the individual with the physical environment, for instance through design concepts such as Design for All. Perhaps what is now needed is a similar design approach relating individuals to their information environment, and the ways in which the physical and information environments can be more effectively integrated at the level of the individual.
In looking at the role of computers in industry it is suggested that it is necessary to take a wider view of their roles as other than simply as device controllers as their use increasingly permeates all aspects of industrial systems. Most importantly perhaps, it could be argued that the relative roles of, and relationships between, humans and computers needs to be reassessed.
This survey article focuses on "*Methods and Applications for the Design, Modelling, Simulation and Integration of Cyber Physical Systems*" and is organized as follows:

- Section 1 gives a brief description of the current state and the motivation.
- Section 2 introduces the characteristics of different types of system (e.g. mechatronic systems, Cyber Physical Systems, Systems of systems, Internet of Things).
- Section 3 then provides an illustration of methods in the design processes of Cyber Physical Systems.
- Section 4 then presents a presents a range of case studies from system level view to a component level view.
- Conclusions and future prospects are given in Section 5.

2. The Evolution of Cyber Physical Systems

This chapter provides an overview different types of system and the transition process from mechatronics to CPS and cloud-based (IoT) systems. Mechatronics can be considered an interdisciplinary field of engineering science which aims to integrate and interconnect mechanical engineering, electrical engineering/electronics, control engineering and computer science (also often called information technology) such that their interactions form the basis for the design of a range of products and product types (e.g. [HTF96]. Interaction between product developers from different fields is often hindered by an insufficient understanding across disciplines, and by a lack of shared platforms for the modelling of complex systems.

Since many sub-systems are delivered by external suppliers, there is a need for both horizontal integration within organizations and for vertical integration between sub-system suppliers and the suppliers of the complete systems. In such a context Lee [Lee08] defined CPSs as the "*integration of computation and physical processes*". Embedded computers and networks monitor and control the physical processes usually through feedback loops in which physical processes affect computations and vice versa.

Figure 1 shows the transition process beginning from a mechatronic system to CPS and the IoT in which the horizontal-axis represents the driving forces/necessary technologies and the vertical-axis the level of the system.

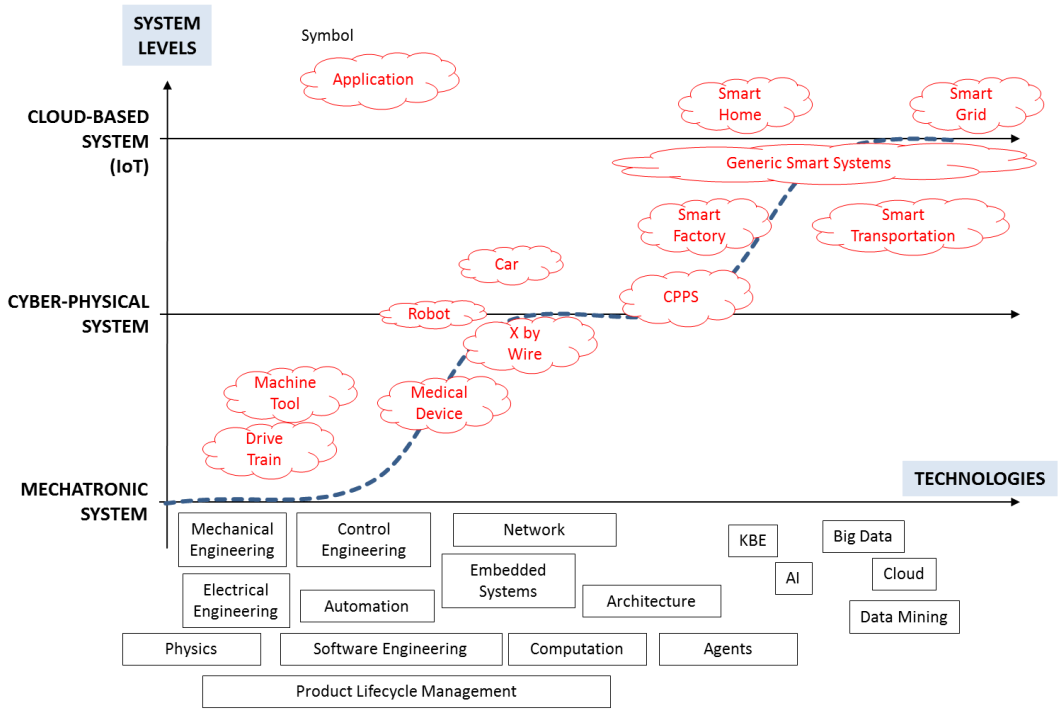

Figure 1: Transition process from Mechatronics to CPS to Internet of Things

2.1. CPS-Paradigm

In the past, the paradigmatic changes associated with the developments in computer technology were not only the drivers of evolution, but also brought about a revolution in systems concepts and thinking [PoHo16]. The promise of the Cyber Physical revolution is that it enables the physical world to be monitored, controlled and influenced both adaptively and intelligently. Many researchers are of the view that the paradigm of Cyber Physical computing will have far reaching consequences in the realm of engineering systems. However, the development of CPSs is interwoven with technological, economic and social opportunities and challenges. There is therefore a need for new transdisciplinary theories, models, methods, tools and contexts to support these developments.

Achieving this capability also raises the need for shared domain semantics and conceptual frameworks in order to create bridges between mechanics, electronics, engineering, control and computation [Lee08]. This is important with a view to the historical development of the current position in which the ontology, epistemology, methodology, axiology, and praxeology of information and computation sciences, on the one hand, and of the physical sciences (natural, chemical, biological) on the other are separated, creating a divergence in both scientific foundations and technologies.

While networked personal computing, embedded, ubiquitous, and pervasive computing all lead to a functional and implementational diversification of types of technical systems, Cyber Physical computing stimulates a technological homogenization and convergence in system manifestation [PoHo16]. In this context, currently differentiated systems are likely to show more and more similarities in terms of enabling technologies and functional spectrum, and this convergence is likely to become increasingly significant in the near future. We are already witnessing a strong convergence of the cyber world (created by human knowledge and digital computing) and the physical world (e.g. in the form of engineered technical systems).

For these reasons, Cyber Physical Systems matter not only from an academic perspective, but also from a governmental and strategic perspective. For instance, the Advanced Research and Technology for Embedded Intelligence and Systems (ARTEMIS) program of the European Union invested seven billion euros in R&D between 2007 and 2013 to stimulate work towards achieving leadership in intelligent electronic systems by 2016. In the US, the President’s Council of Advisors on Science and Technology (PCAST) positioned CPS research as the number one priority for federal investment. The National Science Foundation (NSF) in the US and the Horizon 2020 framework program in the EU have formulated and financed projects working on the theories, technologies and implementation of CPSs since 2007 and 2010, respectively. In 2011, the German National Academy of Science and Engineering (Acatech) launched an integrated research agenda for Cyber Physical Systems [Aca12].

Additionally, many large and international companies see CPS as a paradigm for their future systems and services. In the US, the Industrial Cyber Physical Systems Center (iCyPhy) has been established by a consortium including UC Berkeley and Caltech along with several partner companies. In Europe, a public-private partnership has been established under the title "*Electronic Components and Systems for European Leadership*" (ECSEL).

2.2. The Internet of Things

The underlying concept of the Internet of Things as a medium for the exchange of information between numbers of interconnected devices and systems goes back beyond the point that the name was first coined. What has driven the growth of the concept in recent years has been the increasing availability of smart devices that can connect to the internet and, often autonomously, exchange information between themselves. This has in turn led to the development of new system concepts in which such smart devices are integrated through access to internet based resources (Software as a Service (SaaS), Platform as a Service (PaaS) & Infrastructure as a Service (IaaS)) to generate context based systems on demand. Further, it creates an environment in which information can be seen as a commodity whose value to the system, and hence to the user, is largely determined by context.

The linking of this information, and hence knowledge, based environment to the ability to analyses large quantities of data is creating new business models (Figure 2) whilst creating issues of privacy and the management of, and access to, individual data [BHV14, Eva11, Wor15].

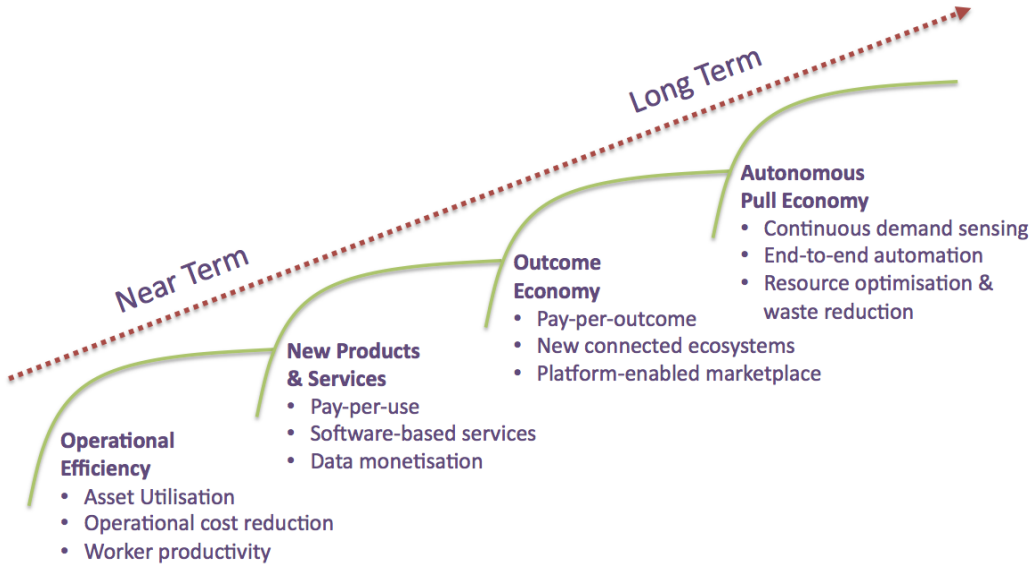

Figure 2: The Industrial Internet of Things (after [Wor15])

Taken together, the above presents a number of challenges for system designers, including:

Nature of Information

- Historically, information has been used to link together artefacts to create a system.

- In the context of the Internet of Things, and indeed Cyber Physical Systems, smart artefacts are used to service information to create a system.
- The information required may be either or both of local or remote and is linked to user need.
- This requires systems which are context aware in order to seek out the information associated with current need.
- Smart modules may be added or subtracted as required as part of a continuum of information and objects.
- Systems negotiate for information based on context.
- Information as a commodity.
- Values are set by user requirement and need.

Uncertainty in Design

This results from the fact that IoT based systems are increasingly self-organising and context based and brings with it a number of issues including:

- The need to ensure effective communication between members of the design team and the means of achieving this.
- Developing and implementing smart, adaptive and user oriented interfaces.
- The modelling of abstractions, particularly during design.
- The effective integration of the information and physical models to coordinate actions, in real time, between both environments.
- Prototyping and testing in the information environment.
- Data security and security of access.
- Social issues associated with the management and use of data, particularly personal data.
- Achieving Privacy by Design in the presence of undefined system structures.

The concepts of the Internet of Things and Cyber Physical Systems presents designers with the challenge of implementing structures within information rich environments where information and communications are increasingly the drivers of the design process. This in turn requires designers to have access to new and novel means of simulation capable of representing such situations.

In the context of industry, the ability to generate new products and services on the back of the information exchange provided by the Internet of Things is likely to be crucial, as is the ability to provide users with the range of smart objects they require to support the resulting systems. For instance, many of the resulting systems, and their associated smart objects, will not be programmed in any conventional sense but will have their functionality specified by users in a variety of ways.

This then requires the interface to have an ability to translate user intent, particularly from non-technical users, into the necessary software structures to define system operation. System operation might in turn require the identification of an IoT based resource, for instance to analyse derived data, and this must be done in such a way that the user system, and associated user data, is not compromised. The implication from the perspective of industry is that there

will be much more uncertainty as to how their product or process is to be used, and how it will be required to integrate with other products and system artefacts from across a range of suppliers to produce new systems.

Referring to Figure 3, the relationship between the various system layers of mechatronics, Cyber Physical systems and the Internet of Things can be expressed in terms of increasing levels of abstraction, to which must then be added the user.

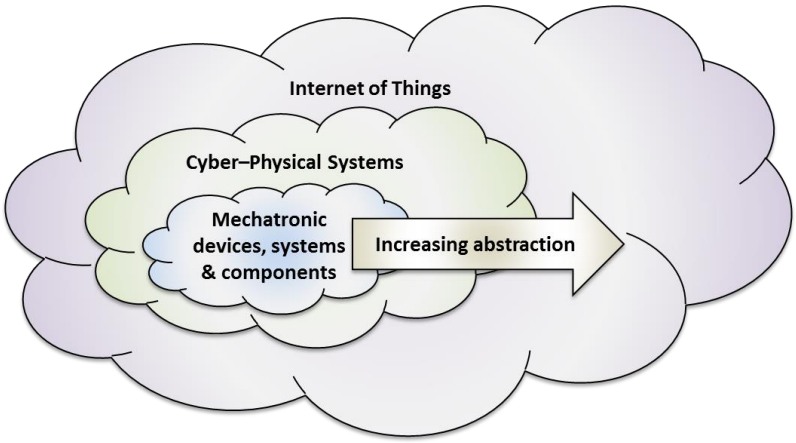

Figure 3: Relationships

Again referring to Figure 3, at the device (mechatronic) level, the design processes and procedures are reasonably well defined and understood and support a translation into physical systems. At the level of Cyber Physical Systems, the relationship between the physical components and the cyber level is largely defined by function. Nevertheless, as systems at this level become more complex, the ability of individuals to understand their operation diminishes. Finally, at the level of the IoT, there are a lot of ill- or undefined constituents which will be unknown to the system builder other than in terms of their desired contribution to system operation and function.

3. Methodologies for CPS-Design

As previously depicted, CPS-design urgently requires the introduction of high-level integrated design methods in which the designers consider all engineering disciplines simultaneously, along with operational issues such as privacy. Early design phases, such as conceptual design and system modelling, play important roles in design, and are considered as the enabling foundations for the final system form. [Wan02, KoTo12]. For the sake of clarity and referring to Figure 4, multidisciplinary integration is discussed from the point of view of both spatial dimensions (disciplines as represented by the horizontal axis of Figure 4) and the temporal dimension (the design process as represented by the vertical axis of Figure 4).

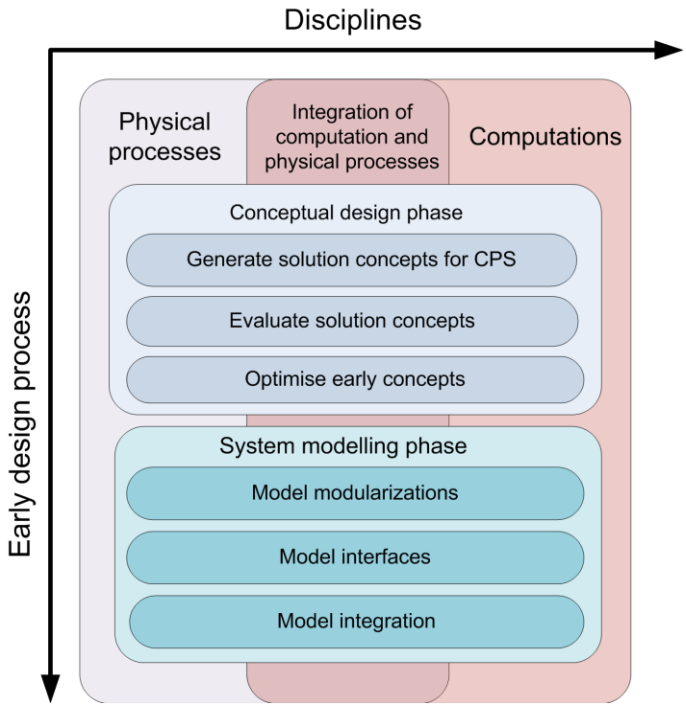

Figure 4: Disciplinary integration during early design process of CPSs [ZLH+16]

3.1. Main challenges of designing and integrating CPSs

CPS-design requires a multi-disciplinary development process [LCB+14] during which designers should focus not only on the separate physical and computational components, but also on their integration and interaction. Challenges related to CPS-design are therefore considered here from the perspectives of the physical process, computation and integration respectively.

- From the perspective of the physical process, CPSs link directly with the physical world. They detect changes in and to this world and realise the real-time control of

system behaviour. Compared with other more conventional forms of complex system (e.g. mechatronic systems), environmental influences (i.e. variances and uncertainties in environment or created by human beings) play a significant role for CPS [GAW14].

- From a computation perspective, CPSs are considered to be developed from and based on embedded systems [LeSe11]. However, the link to physical world of CPSs is much stronger than that of embedded systems. Therefore, the software component of a CPS is much larger and more complex than that of an embedded system.
- From the perspective of the integration of the computation and physical processes, CPS-design requires a integration within the collaborative design process of CPS. Physical and computational components interact with each other between disciplines, leading to interruptions and incompatibilities during the design process. There is also often a lack of clearly specified and documented interactions and interfaces between the various disciplines and involved components and hence mutual understanding in communication is hindered [GAW14].

3.2. Functional and architecture design of CPSs

Current research together with prior experience suggests that there is a lack of tools and methodological support available to handle the complexity of CPS-design in the early design phases [APA13]. This is in turn directly linked to the functional and architecture design of CPSs. From the perspective of the conceptual design phase, the design-characteristic aspects of systems, such as hierarchy parameters and modularity should be analysed using a conceptual model [Heh12]. The solution concepts for CPS-design should then be proposed at the end of the conceptual design phase. Three main challenges exist in the conceptual design phase as follows.

- Firstly, how to generate solution concepts based on customer requirements or market survey results? Although various tools and languages have been proposed to structure customer requirements [INC10], the process from requirements specification to solution concepts should be defined. Different solution concepts can be proposed by individual designers, or design groups, according to their background knowledge and professional experience.
- Secondly, how to evaluate the different solution concepts proposed.
- Finally, because of the complexity of a CPS, it is impossible to find the optimum solution concepts without iteration and there is a requirement to optimise solution concepts during the early part of the design process.

The Systems Modelling Language (SysML), has been established by the International Council of Systems Engineering (INCOSE) and the Object Management Group (OMG) [OMG09] based on unified modelling language (UML) to support Model Based Systems Engineering (MBSE). Other modelling languages have been proposed to model a CPS.

Venter and Ehlers [VeEh12] believe that it is not possible to formally describe dynamic a CPS due to the flexibility in which incomplete UML diagrams can be defined, so they propose an Architecture Description Language (ADL) for dynamically modelling architectures with specific reference to CPSs. A highly dynamic and distributed autonomous architecture can then be described by the ADL. Unlike modelling languages such as UML/SysML or ADL, Modelica can not only represent a system, but the model created in Modelica can also be transferred into mathematical models for simulation solving. Several design methods based on Modelica have be described [HeEl11, Fri11, FeZh14, NSE15].

3.3. Methods and tools for the modelling and simulation of CPSs

From the perspective of the system modelling phase, a CPS may be defined as an assembly of components and the associated interfaces between them. The objective of the system modelling phase is to set up a cross-multidiscipline model to determine the important functional and system parameters and hence realise the potential optimisation [Hehe12]. The cross-multidiscipline models created during the modelling phase of the CPS-design process are hindered by three main issues. According to the definition of a CPS, models of a CPS normally include both physical and computing components [LeSe15], so first concern is as to how to model the components of the different disciplines. The second issue is that of how to model the interfaces among components drawn from different disciplines with a common terminology. The final issue is related to model integration. Established techniques exist to model the dynamics of the physical components (i.e., a continuous model) and the discrete behaviours of the computing components (i.e., the discrete model), but the interface to join the continuous model and discrete model has not to date been given significant attention.

Petnga and Austin [PeAu13] have proposed temporal semantics and their central role in the development of a new time-based reasoning framework within MBSE for CPS. In their proposition, an ontological framework describes the system behaviour in terms of time, intervals of time, and relationships among intervals of time, and the correctness of functionality with respect to requirements can then be assured. Lin et al [LSM10] developed an agent-based modelling approach for CPSs to help designers solve the problems resulting from the fundamental differences in the operation of the cyber and physical components as well as from the significant interdependencies among these components.

Besides design methods based on modelling languages and ontological modelling approaches, several design methods based on commercial multi-discipline modelling and simulation software, such as MATLAB [AlAh14], AMESim [CAR14], have been developed to support CPS-design based on MBSE. The modelling software provides various types of libraries across different disciplines to help designers validate, evaluate and optimize their design.

3.4. Product Lifecycle Management for CPS

Various research projects have been proposed relating to the specification of a PLM approach for CPS which supports the integration of a networked economy, the growth of the Internet of Things and the development of mechatronic systems [SRF+05, BTR+14]. Considering the emergence of the CPS concept there is a need to develop product models and systems engineering methods from the mechatronics field towards that of CPS [ZBL+14, BLE16]. To develop such complex systems as represented by CPS, companies currently use several kind of information systems (IS) to manage and integrate the engineering data related to each of the disciplines involved during the design project.

For instance, Product Data Management (PDM) systems are generally used to manage hardware (HW) data [EGN+04], whereas Software Configuration Management (SCM) systems are used to manage software (SW) data [BTR+14]. PDM is generally considered to be one of the most important IS building-blocks for implementing a Product Lifecycle Management (PLM) approach. PLM is defined as a systematic concept for the integrated management of all product- and process-related information throughout the entire lifecycle, from initial idea to end of life [TBD+10]. Focusing on the basic functionalities (version control, concurrent development, configuration selection and workspace management), SCM is very similar to PDM, as Application Lifecycle Management (ALM) can be compared to PLM. Indeed, ALM "*has emerged to indicate the coordination of activities and the management of artefacts* (e.g., requirements, source code, test cases) *during the software product's lifecycle*." [KäVä09] ALM originates from the Configuration Management (CM) domain, which traditionally provides storage, versioning and traceability between artefacts [DoCh11]. It has extra functionality to support communication, reporting, traceability and tool integration, such as requirements and defects management or build and test facilities, as illustrated by Figure 5.

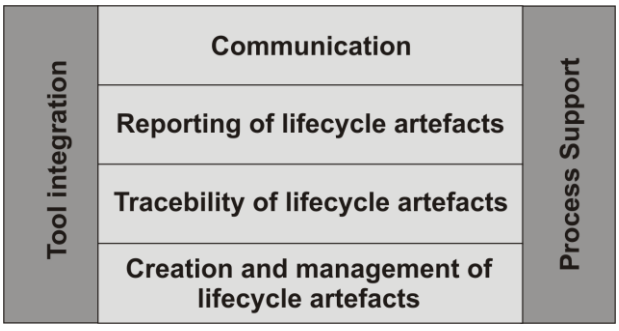

Figure 5: Main elements of Application Lifecycle Management [Kääriäinen et al. 2009]

Even if ALM and PLM are converging in a certain number of aspects (e.g. to take into consideration several life cycle phases), challenges remain for engineering data management during the design of mechatronic systems or CPSs. For instance, having a unified Bill of Materials (BOM) that combines all HW and SW components and defining a common data model that manages the entire system in each design phase are open issues, as shown in

[BLE16]. This unique BOM is essential to enable tracking the design changes and detecting the related impacts, especially between different disciplines. But having a unique BOM, splitting up the mechanical parts, the electronic/electrical components and the software codes is not enough. [ŞeCa12] explain that the same aircraft equipment (i.e. subsystem) can be used with different software on different aircraft. In that case, equipment and software have to be managed accordingly.

To insure that design changes performed on equipment or on software do not impact on the global consistency of the system, the versioning and the configuration selection functionalities of ALM and PLM should be adapted to the chosen product structure [BLE16]. This conclusion leads to another important issue regarding the interoperability of information systems and engineering applications [RSB+08, BTE+10, EGV+11, EGT+15]. The integration of services into product development and the close integration of product development and development of production systems (CPPS) is discussed in Kernschmidt et al. [KVM+13]. Additionally the cross-disciplinary character of such systems and potential inconsistencies during all design phases are proposed by Feldmann et al. [FHK+15].

3.5. Architecture and behaviour paradigm for CPS

3.5.1. System Architecture

The current trend of CPSs of more sensors, more actuators and more connectedness, will not stop. A future CPS might consist of literally millions of sensors and actuators controlled by thousands of CPUs. Having so many active and passive elements in an ever-changing environment (as the system becomes bigger, the territory the system covers becomes bigger, so there will be increasing numbers of changes and fluctuations). A simple calculation tells us that the reliability of the entire system will become the single most threatening issue [Kol14]. This means that the system needs to flexibly adapt to an ever-changing external environment (e.g., disruption) as well as to internal conditions (e.g., failures). In addition, future CPSs need to operate autonomously, i.e., human-intervention free. In particular, failures of the system need to be diagnosed and repaired without immediate human intervention. The concept of resilient architecture is considered promising for these reasons.

The minimal architecture of a CPS consists of three elements; sensors, actuators and intelligence (controllers), what differentiates a CPS from a mechatronic system is the intelligent interconnection and information sharing between devices. Systems architecture dictates how sensors and actuators are organised and behave in responding to the external environment and the internal conditions defined by the controllers [ISO/IEC/IEEE42010]. Figure 6 depicts a simple current CPS architecture. While within a machine, sensors and actuators are connected through a bus (e.g., Profibus) and controlled by the controllers, these machines and robots will be further connected to a network (e.g., Profinet, so called network connection in Figure 6 green arrows) and controlled by a higher level controller. Obviously, this architecture allows for reconfiguration at the network level but not at the machine level.

In contrast, future CPSs built with resilient architectures should allow both static and dynamic reconfiguration at any level [SBD+09]. Such a resilient architecture can be achieved by virtualisation, which means that the logical function of, e.g. the sensor network, will be virtually defined and separated from the physical network [TBK+13]. If there is any failure detected within the sensor network or if the sensing purposes change, sensors will be logically reconfigured through reasoning at the architectural level. Similarly, failures of actuators and controllers can be detected and accommodated through functional reconfiguration.

Figure 7 illustrates an example implementation of such a resilient CPS architecture which allows for the logical reconfiguration of any element. In relation to Figure 7, sensor networks and actuator networks will be virtualised. Thus, every time the purpose of the system, the external environment or the internal conditions changes, the entire system will be able to optimally reconfigure its sensors and actuators based on the reasoning of the integrated intelligence (controller). For instance, in Figure 7, Controller 1 uses Sensor 1 to control Actuator 1. Suppose Actuator 1 is found to be faulty. In that case, Actuator 2 with a similar functionality can be engaged using Sensor 2 (instead of Sensor 1). This “reconfiguration” facilitates the resilience of the entire system.

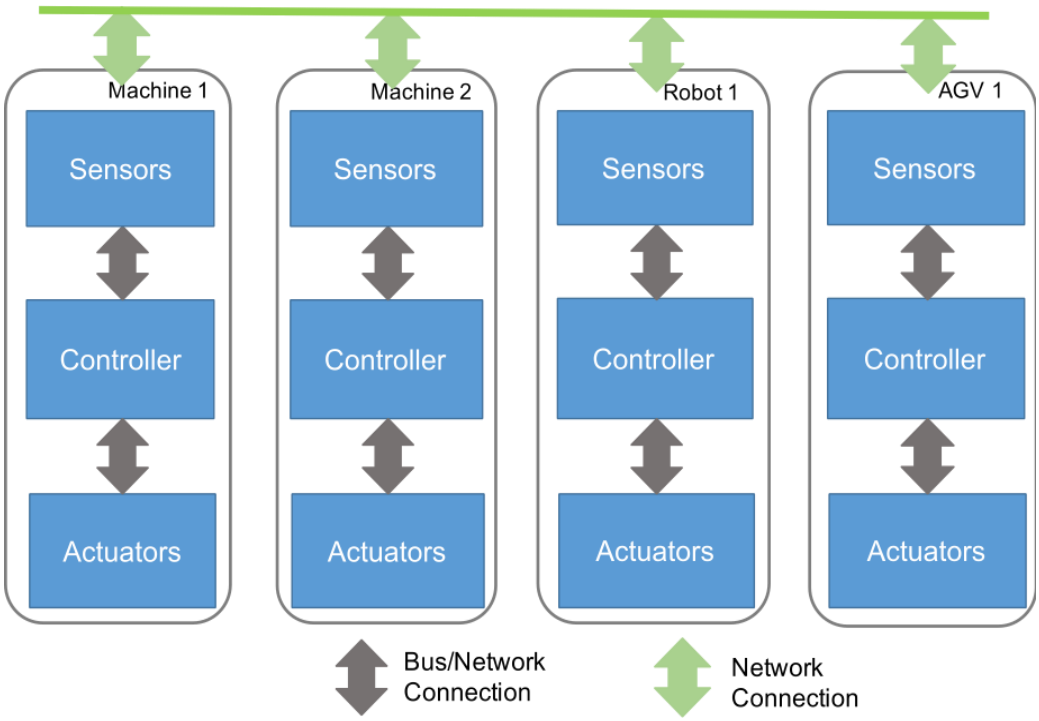

Figure 6: Current CPS Architecture

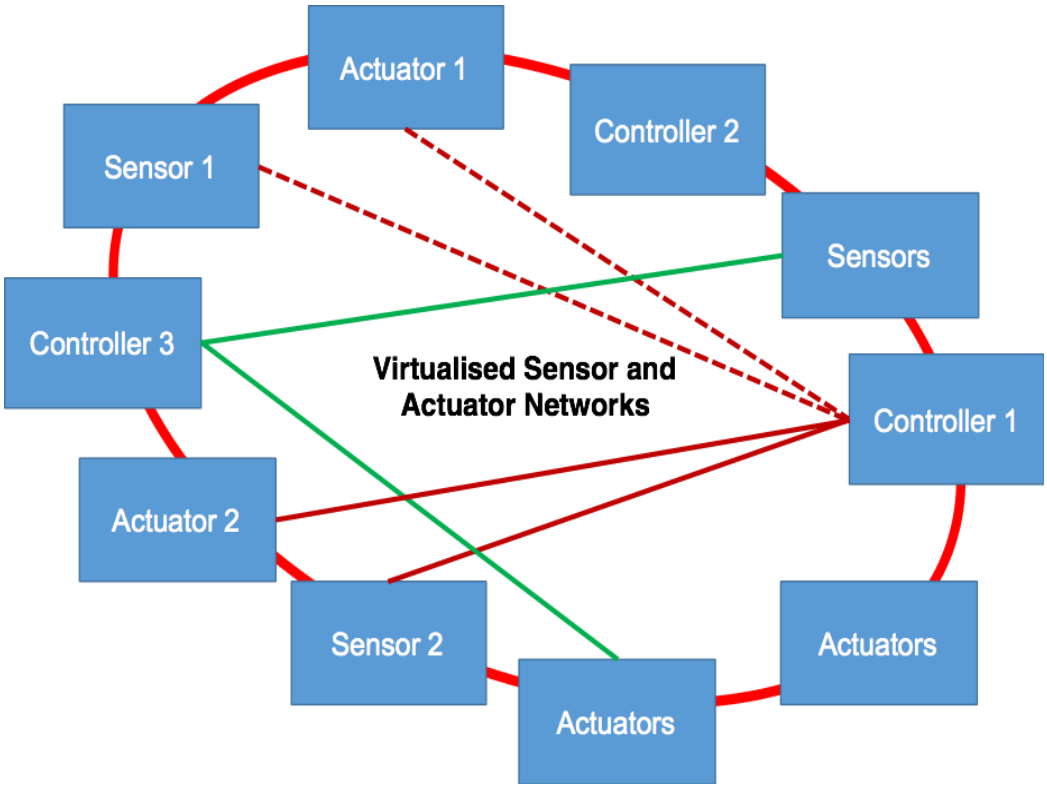

Figure 7: Resilient CPS Architecture

Although the sensor network can incorporate redundancy to increase its reliability, this may increase the uncertainty of the sensory data [HeSo09, RGL+12]. Similarly, the virtual sensor network composed of sensors that are not originally meant to sense the concerned physical

quantity may also introduce inaccuracy and uncertainty. Similar to sensors, redundant actuators can be useful to support the functionally reconfiguring of the actuator network when some actuators are faulty or to exhibit different functions [JoFo13]. For example, an electric car equipped with an in-wheel motor for every individual wheel (thus four motors in total instead of one) can be steered by controlling the speed of the four wheels individually. These motors can also replace mechanical brakes (by using regeneration and by generating a counter-rotating torque), which means that a "*virtual brake*" that could back up the mechanical brake is now available. This systems level architectural design approach to virtual actuator networks has the potential not only to provide better fault tolerance, but also to generate a completely novel and innovative solution. The dynamic reconfiguration and resilience of virtual actuators is a key to achieving the resilient behaviour of the system.
Virtual sensors and actuators networks need to be developed and reconfigured through systems level reasoning about the physical sensors and actuators. These reasoning capabilities play a central role in the entire systems architecture in order for it to become resilient to external changes and failures.

3.5.2. Agents in Architecture Design

In recent years, agents and agent orientation have reached the level of a new architecture and behaviour paradigm. Initially emerging from the area of artificial intelligence [Wei99], they have been integrated within wide areas of the control of production systems [LMV13, VDI10, SHY+06, MVL11, Wag03, VGL15]. An agent in the context of production automation can be seen as a defined entity intended to reach its objectives independently but cooperatively, while inter-acting with its environment, which may include other agents [VDI10].
The agent oriented approach has become a suitable and efficient means for the solution of complex problems within engineering and control systems for production systems. This approach gives the real and existing entities of a production system a central role and, in addition, explicitly considers the purposes and aims of the production and the necessary behaviours to achieve them. Thus, the agent oriented approach is especially applicable to problem classes with complex and highly volatile border conditions as well as the problem of self-adaptability. Examples of such system classes are production planning systems, logistic systems or maintenance systems [LMV13].
As mentioned above, agents may be implemented on PLCs and PCs depending on the real-time requirements and processing power needed. In case real time is required PLCs are chosen here [USV14]. Dependability is guaranteed using a safety shell that allows only safe and non-destructive decisions of the agent to be conducted [WaVo08] during runtime. The agents' knowledge has instead been modelled [VGL15] during engineering as Parameter Diagram, but may also be learned during runtime. An example of the learning behaviour of a small laboratory plant is discussed by Folmer et al. [FoVo13]. Additionally, agents are beneficial in detecting patterns in operation, e.g. alarm patterns in order to reduce the floods of notifications significantly (Figure 8).

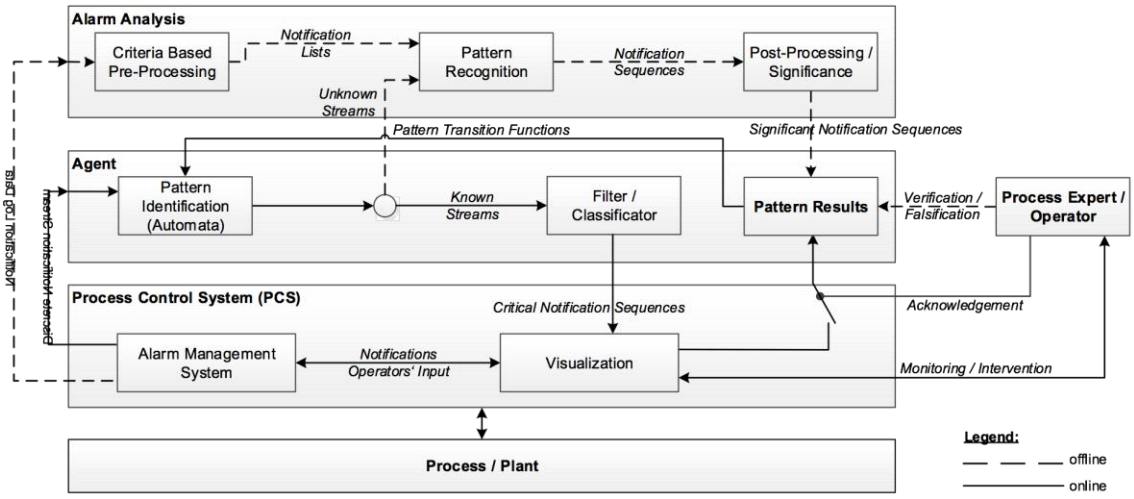

Figure 8: Building blocks for intelligent, dynamic alarm management using agent oriented pattern identification [VLL15]

The alarm management agent introduced by Vogel-Heuser et al. [VLL15] filters and aggregates notifications, forming the agent's knowledge base on a finite automaton. The discrete notification streams of the operating plant are analysed and filters and classifiers are used to further reduce the number of relevant notification sequences and indicate the root cause of the abnormal situation. Finally, solely critical situations, i.e. critical notification sequences are grouped and visualized to the operator to facilitate them in reacting to alarms by reducing the number of notifications, and consequently the associated management effort, significantly. This approach will be discussed in more detail in the following paragraphs.

The approach is based on a concept for the frequent pattern mining of industrial notification logs to identify pattern sequences comprising causally dependent notifications. The algorithm which implements the proposed concept is designed to provide a two part analysis. As the first part it applies abstract criteria-based recognition strategies (RS) designed to combine tags commonly configured in alarm management systems. These are not pre-defining notification patterns that should be identified, instead, they define strategies to find correlated sequences based on abstract a priori knowledge by taking configured alarm tags into account. In the second part, the frequency pattern mining algorithm analyses the notification logs to identify frequently occurring patterns when the RS is applied on identified patterns. Applying the implemented pattern recognition, the algorithm is able to generate formal rules that are transferable to a graphical representation for reviewing detected sequences that are human readable. This greatly increases the comprehensibility of emerging sub-sequences for human users.

A critical component is the agent's knowledge base. The agent has to visualize all critical situations dependably while restrictively filtering and aggregating the notifications. A human in the loop machine learning approach was chosen to enable the automatic detection of notification sequences while being supervised by humans. Historic alarm logs are used to

identify significant notification sequences based on statistical pattern recognition techniques to suppress redundant notifications and visualize the critical situations [FoVo13]. As has been identified [FoVo12], applying machine learning to historic alarm data would result in notification sequences which are statistically correct but logically nonsense. In order to reduce operator effort, criteria based pre-processing grounded on background knowledge like reference designation and plant documentation is applied. Additionally, post-processing based on significance tests is applied to further reduce the number of patterns to be reviewed by the operator.

Beyond some initially promising research and development activities [TVK09], the required real-time capabilities of agent technology, which limit the applicability of agent technology, are still an open problem. Nevertheless, beyond agent technology as software technology, agent orientation as a modelling and engineering paradigm is currently completely applicable. This is especially valid for use cases dealing with limited hard real-time conditions such as in cases of manufacturing execution control, maintenance, data allocation, and documentation.

4. Applications, Industrial Practice and Issues

This section presents illustrative case studies and current best practice from industry. The examples are selected from different system levels starting with the description of the overlaying concept of Cyber Physical Production Systems (CPPSs). Afterwards the model-driven engineering process of a self-configuring agent-based CPPS will then be discussed. The analysis and evaluation of specific properties of a sub-system using a condition monitoring system, important for the maintenance purposes, is given for a wind turbine. This leads to the potential implementation of new maintenance concepts for Cyber Physical Systems such as "*robotic maintenance*" and "*autonomous maintenance*".

4.1. Cyber Physical Production Systems (CPPS)

Cyber Physical Production Systems (CPPS) are Cyber Physical Systems as applied in the domain of manufacturing/ production, in Germany the term "*Industrie 4.0*" is used. There are several definitions of Industrie 4.0 (I4.0), though there is a general agreement the following design principles [Industrie4.0, VoHe16]:

1. Service Orientation - CPPS offering services via the Internet based on a service oriented reference architecture.
2. Intelligent self-organizing CPPS providing:

   The ability of a CPPS to make decisions on its own (decentralization).

   The ability of CPS, humans and CPPS to connect and communicate with each other (interoperability).
3. Information aggregation and representation for the human in the loop during engineering and maintenance of aPS (automated production systems)
4. Virtual copies of a CPPS at different levels of detail, e.g. from sensors and actuators to the entire CPPS (virtualization).
5. Relevant process and engineering information for data analysis (real-time capability).
6. The ability to flexible adapt to changing requirements by replacing or expanding individual modules (cross-disciplinary modularity).
7. Big Data algorithms and technologies provided in real-time (real-time capability).

8. Optimization of the manufacturing process based on these algorithms and data to increase Overall Equipment Effectiveness (OEE).
9. Data integration across disciplines and along the life cycle based on standardized data models and a model driven modular engineering process
10. Secure communication enabling a worldwide network of aPS supporting economic industrial partnership across companies borders,
11. Access to data securely stored in a Cloud or Intranet.

In the following the focus is on requirements 1, 2 and 7. The architectures of industrial automation systems have for many years been dominated by an approach comprising several different layers e.g. for the field-level of production systems, Manufacturing Execution Systems (MES) and up to the Enterprise Resource Planning (ERP) layer. To support technological development, this hierarchical architecture has been adapted to a two-layered architecture, in which intelligent field-devices can be integrated and connected by an information model to different ERP and MES functionalities (see Figure 9). Service oriented and intelligent CPPS provide their status and their technical resources as a basis for their capabilities and define the relevant data points for other CPPSs by a connector, which may be connected via OPC-UA or other form. Agents may help to connect to other connection standards. The resource description, i.e. the hierarchical sub-systems and available behaviour is given in Automation ML (CAEX and PLCOpenXML).

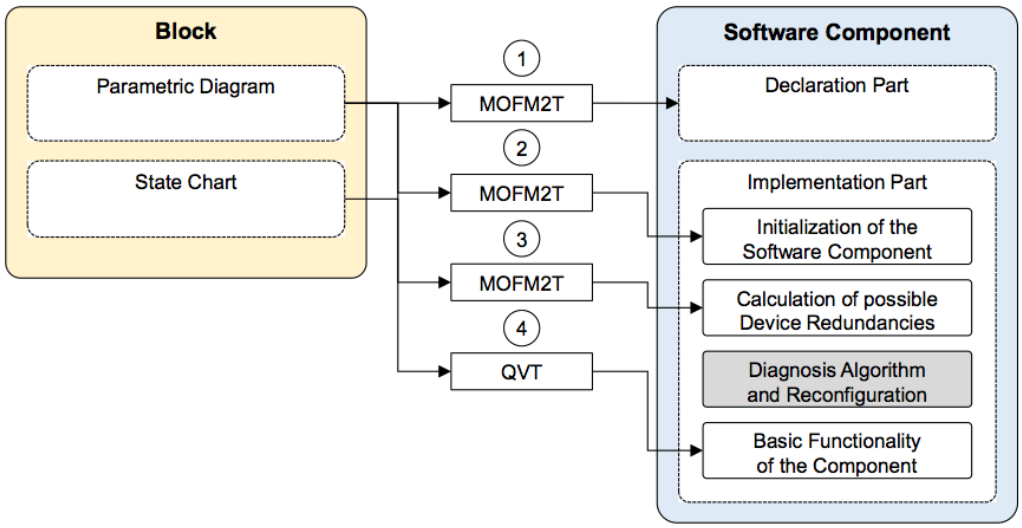

Figure 9: Agent-based self-aware, self-describing CPPS-module

To achieve the required ability to make decisions on their own during runtime, CPPSs need to provide a certain level of intelligence, perhaps realized by implementing this knowledge as agent knowledge and allowing the agents to decide autonomously but after negotiation with their cooperating CPPS and having the ability to learn from their operation. The tank example discussed in the following section shows an example application of such knowledge implemented at the automation level on a Programmable Logic Controller (PLC), as well as the ability of the agents to learn patterns from alarm notifications. One of the key challenges

for CPPS in the near future are to define the metrics for evaluating I4.0 compliance and the benefits of adaptivity, flexibility etc. allowing for the benchmarking of different solutions.

4.2. MDE of Self-configuration agent-based CPPS

To achieve an intelligent reconfigurable CPPS, a model based approach may well be beneficial [Vya11, Thr13, LZV+11, DOB+13, LeRo12, BFS12, CPdA+12, PSL11, LSVH13, SWL+13]. Here we introduce an SysML based approach to developing self-configuration abilities of field level functions using indirect redundant information or devices, i.e. in case of a fault, the system is able to reconfigure and continue operation to avoid down time using already available neighboured values or related values and devices.

In the SysML-based approach, single components of a CPPS are modelled as an agent using the SysML element *Block* to encapsulate an agent's description model. This model covers information that describes the basic functionality a particular software component fulfils within the CPPS, e.g. handling or processing work pieces, using a State Chart and the information on which devices can be replaced by redundant information in case of failures using a Parametric Diagram (cf. Figure 9, left). One of the main benefits of model based approaches is the possibility to automatically generate the designed software components using model transformations. In the presented approach this benefit is used by applying the standard model transformation mechanisms of the OMG (cf. Figure 9, centre), i.e. the MOF Model-To-Text Transformation Language (MOFM2T) and the Query/View/Transformation-Language (QVT). By using these mechanisms, model transformations have been developed that generate the different parts of the software component automatically.

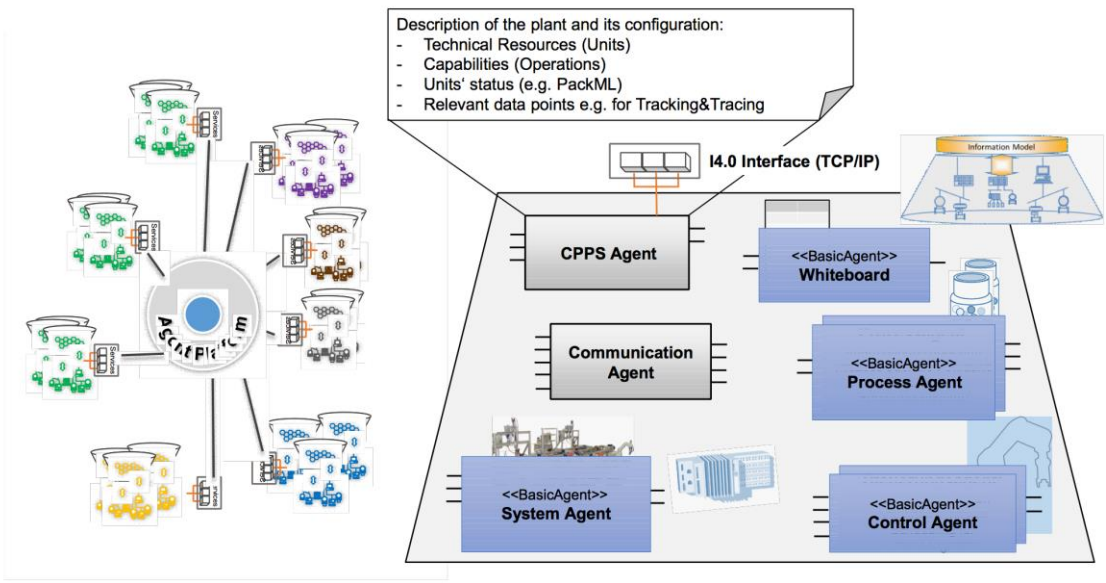

Figure 10: Overview on the SysML based approach to design agent-based self-configurable CPPS [Vog15]

As the Parametric Diagram captures the information on the CPPS component's devices, e.g. sensors, and their relationships to possible redundant information or devices, the declaration

part of the software component is generated in the first step (cf. Figure 9, centre). The implementation code for necessary calculations is also generated from this. Since the diagnosis algorithm has been developed to work independently from the number of devices in a component (cf. [SWL+13]) it does not need to be generated from the SysML model but can be placed inside the software component as static code (cf. Figure 10, right). In the last step, the implementation code that realizes the basic functionality of a CPPS component is generated from its description, e.g. inside a State Chart.

The required SysML model will be discussed in the following paragraphs and illustrated by an application example from process industry. This considers a tank (see Figure 11) equipped with two sensors to detect the maximum and minimum filling levels. The tank is used to store liquids in a production process Once the minimum filling level of the tank is reached (detected by the lower tank sensor, sensor 101.2 in Figure 11), a valve (V101) is opened and the filling process starts. The filling process will run until the upper tank sensor (sensor 101.1) signals that the maximum fill level of the tank is reached. However, there is no redundancy in the process and thus a malfunction of the upper sensor leads to an overflow of the tank.

As a precaution and in order to detect a failure of the upper sensor, the original tank function is enlarged by a self-configuration method according to Gronau's classification of adaptivity [GLA07]. For self-reconfiguration, the fill level of the tank is calculated based on the amount of fluid that is fed into the tank, the amount of fluid that flows out of the tank, the tank's dimensions as along with the filling and emptying times.

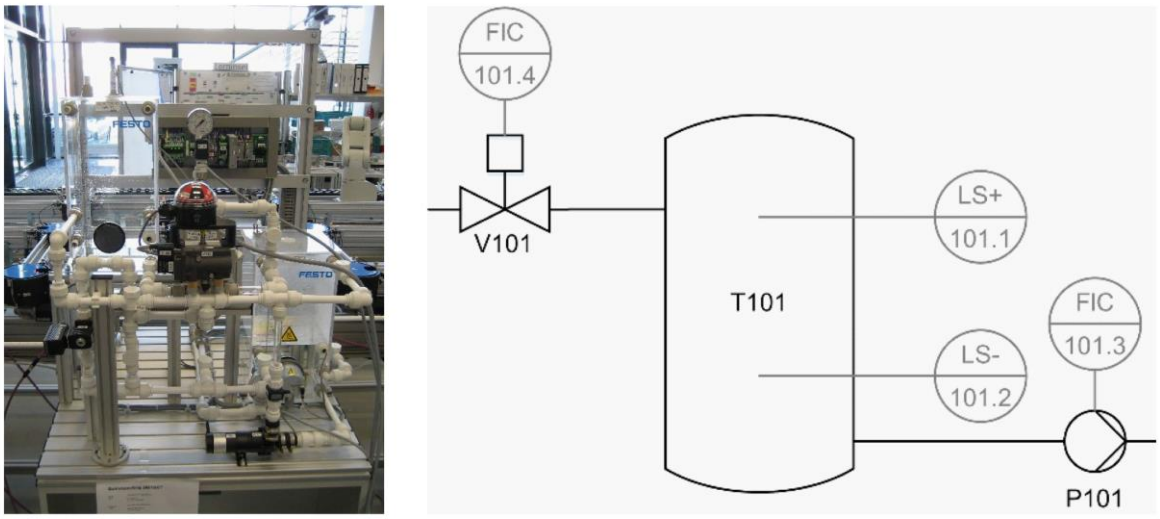

Figure 11: Tank with upper and lower filling level sensors, valve and pump [FPS11]

If the inflow or outflow or the level sensor is faulty, a filling level error and an inflow error and an outflow error would occur. The real level sensor can be approximated by a virtual sensor calculated during runtime. If the filling level sensor is effected by a fault, the faulty situation may be detected. At first the redundancy model has to be developed in the design phase by taking functional dependencies into account and formulated as physical equation (cf. Figure 12, left). To map the redundancy model onto the SysML based notation of the approach using the Parametric Diagram (PAR), each real sensor of the plant (Figure 12, left,

ovals) is modelled as a Block inside the PAR. For the physical equations which represent the functional dependency between the sensor values a Block has also to be modelled inside the PAR. In the code generation process, both types of Block are translated into specific sections of the generated software component that either handle the pre-processing of real sensor values or calculate redundant sensor values based on the modelled physical equations.

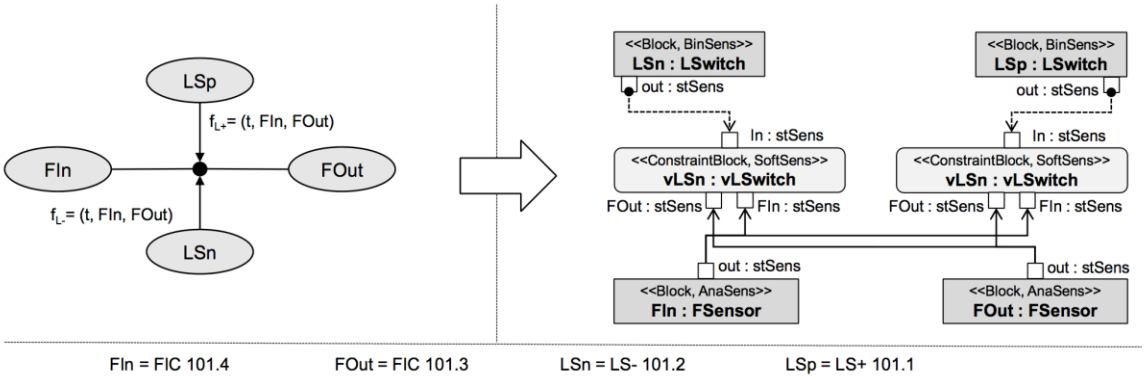

Figure 12: Redundancy Model according to [SWL+13] (left) and mapping to PAR (right)

Since the redundancy model has to be implemented on the PLC to allow decisions during runtime, the code generation of the SysML based approach supports (cf. Figure 9) the transformation from the information modelled within the PAR into software components according to IEC 61131-3, the leading standard for PLC programming and the standard hardware platform for automated Production systems.

As the basic step of the code generation process an IEC 61131-3 Function Block (FB) is created as from the agent modelled in SysML for the software component that is implemented inside the CPPS (Figure 9, right). The process of the code generation consists of four further steps to completing the generation of the FB (cf. Figure 9, middle). As the first step the necessary declarations for the virtual sensors and real sensors are generated and a redundancy matrix created as a code variable array inside the declaration part of the FB. In the second step, a code section inside the implementation part of the FB is generated which initializes the entries of the redundancy matrix.

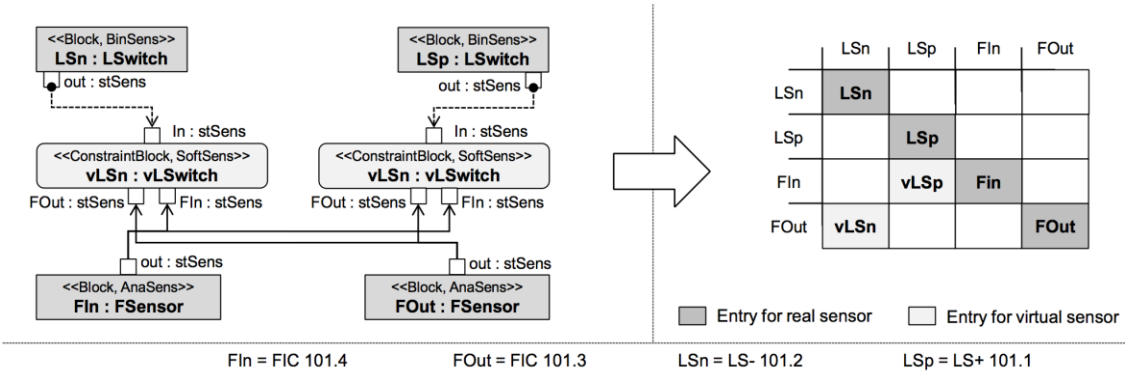

Figure 13: Mapping of PAR onto Initialization of the Redundancy Matrix [enlargement of SWL+13 and FPS11]

Each cell of the matrix represents a relationship between two sensors. If a relation exists (modelled as a virtual sensors inside the PAR, see Figure 13), the cell is filled with the soft sensor value calculated on the basis of the analytical dependency. The columns and rows are labelled with references of all nodes. The main diagonal is periodically (each PLC cycle) updated with the values and qualities of real measurements. Replacements for a sensor are available in the same row. Transforming the knowledge base in such a matrix allows for the reuse of the same algorithm developed in previous work to compare the available values at a measurement point and detect a failure as well as to choose the best value currently available. As the second step of the code generation process, the calculation of the analytic dependencies are generated from the PAR. Therefore, the blocks modelled for real sensors and virtual sensors are put into the sequence at the point at which they need to be invoked and implementation code is generated in the implementation part of the FB in that sequence. Finally, for the generation of the basic functionality of a CPPS component, further previous work to develop code generators to translate State Charts into executable IEC 61131-3 code is adopted. More detailed information on the code generation from Parametric Diagrams can be found in [VHSL+14] and on the code generation from State Charts in [WiVo08].

4.3. Wind Turbine Condition Monitoring

Wind turbines (see Figure 14) convert kinetic energy from the wind into mechanical energy driven by rotors and control technology. In past decades, the installed capacity of wind energy converters worldwide grew exponentially as a result of system cost reduction. To keep the cost of energy down and increase profit margins, operators need to be able to monitor the performances of their turbines more accurately [SSA13a, SSA13b, WMM+11]. The basic idea of all model based approaches is to identify closely related signals in order to use them to build a prediction model of normal versus abnormal behaviour of the system [SSA13a]. However, wind turbines do not work alone; they generally work in clusters which are referred to as wind turbines farms. In order to monitor the condition of these farms, industrials mainly use treatment of data obtained from the Wind Turbine Supervisory Control and Data Acquisition (SCADA data). The data treatment is not performed on site but generally in the parent company by the engineers who would remotely send instructions for repair, reset or shutdown (and in extreme cases move physically in-situ); these interconnected actions create a cluster of interconnected objects including the wind turbines, the control devices and the software condition monitoring solution. These interconnected objects form a Cyber Physical System (CPS) [AcTo16].

SCADA data from wind turbines farms tend to be very large and therefore they set new challenges in data analysis where traditional statistical methods break down partly because of the increase in the number of variables associated with each observation [Süt10, ScSa10]. While certain computational methods can construct predictive models with high accuracy from high-dimensional data, it is still of interest in many applications to reduce the dimension of the data sets prior to any modelling mainly through feature selection. In the next sections the basic steps for construction a condition monitoring system for a wind turbine CPS are considered.

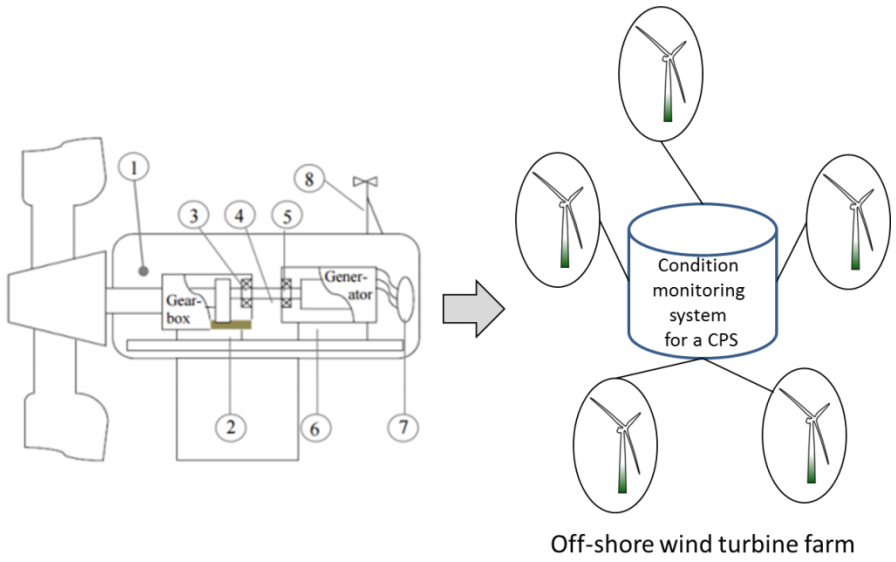

Figure 14: Wind Turbine with Embeded Signals [Süt10]: 1: Nacelle temperature (To) 2: Gearbox oil To 3: Gearbox bearing To 4: High speed shaft revolution speed 5: Generator bearing To 6: Stator To (3 phases) 7: Power output as a part of a Off-shore wind turbine farm

Firstly, it is worth noting that the SCADA data used in this study are collected over a one year period, from 9 off-shore 2MW turbines. The SCADA data represents information from 51 measured input and output signals (every ten minutes). It would be numerically expensive and unpractical to create a prediction model with all the inputs. Therefore, for feature selection genetic algorithm (GA) is used combined with a Partial Least Squares (PLS) approach to automatically reduce the number of variables (features) to represent the CPS dynamical behaviour.

A GA has three basic operations; reproduction, crossover and mutation [Gol89], yielding positive results in a number of practical applications [AcAp10]. PLS is a statistical method used for extracting relevant parts of the information from data sets. By combining the two (GA-PLS), the selection of variables and the performance prediction of PLS models correspond to the population strings and fitness function of the GA. The GA-PLS presented by Leardi & González [LeGo98] is used with the following setting parameters: size of the chromosome = 30, Mutation Probability = 1% and Cross-Over Probability = 50%. Additionally, and based on previous studies, the models will be exemplified through Hydraulic Oil Temperature monitoring (output).

Once the GA-PLS is applied to the SCADA data, and with fine-tuning of the setting parameters, we obtain the most important variables to predict the Hydraulic Oil Temperature behaviour of the wind turbines. The bars in Figure 15 represent the amount of times the GA-PLS extracts the variables to model the target. The five (5) variables appearing most frequently are the: Generator RPM Average, Generator Bearing Averaged Temperature, Generator Phase Temperature, Gear Bearing Temperature Average and Ambient Wind Speed Average.

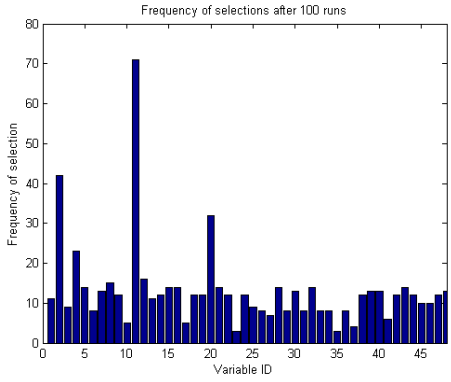

Figure 15: GA-PLS bar chart illustrating the importance of the variables regarding the output

4.3.1. Model Prediction Using ANFIS

Once the variables selected, we need to develop a model that mimics the behaviour of the data set. For this purposes and based on previous work [SSA13a, ASR+16] we opted to use the adaptive network-based fuzzy inference system (ANFIS) tool, an artificial neural network based on the Takagi–Sugeno fuzzy inference system. The Root Mean Square (RMS) error is

used by ANFIS to compare the differences between the predicted values and the actual measurements in the data set and is calculated as:

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(y_{measurement} - y_{predict})^2}{n}}$$

where $y_{measurement}$ is real output values and $y_{predict}$ is output values from ANFIS.

4.3.2. Membership Function Type

In ANFIS one can select several types of membership functions (MF) as in Triangular, Trapezoidal, Gaussian, Gaussian 2 and Generalized Bell Curve. All of which were tested to evaluate their predictive capabilities on the learning data set that was built from the available data. For the comparison we used 2 MF and 4 MF per variable. The number of generations (epoch) was set to 10 as shown in table 1.

Table 1: RMS-error for different types and # of MF by grid partitioning.

#epoch=10	**Triangular**	**Trapezoidal**	**Gaussian**	**Gaussian 2**	**Generalized B.**
2 MF	1.40	1.41	1.35	1.36	1.35
4 MF	1.24	1.45	1.19	1.39	1.45

In our case, the Gaussian membership function gives the best results across both tests. The ANFIS model behaves better with 4 Membership functions per variable, however this increases the number of rules from 25 (32 rules) to 45 (1024), which in turn increases the model learning time from around a few hours to 4 days which is not worth the gain in precision. Now let us investigate the distribution of MF using data organizing methods.

4.3.3. Membership Function Distribution

ANFIS tool is capable of organizing the data for training using Grid Partitioning (GPart), (used to determine the type of MF) and two other clustering methods, Fuzzy C-Means (FCM) and Subtractive Clustering (SC). The RMS-error and execution time are used to determine, how well the different data organizing methods perform. The RMS-errors for the three types of data organizing methods are listed in table 2.

Table 2: RMS-error for gearbox bearing temperature for different data organizing methods (# rules =32).

	GPart	**SC**	**FCM**
#Epoch = 10	1.35	1.78	1.39
Time	04:35:07	02:39:01	06:34:02

One can see here that GPart performs the best. The SC performs the worst, but was the fastest by far. A choice can be made by the operator, since the error difference is not that large, we chose to go forward with the SC.

4.3.4. Prediction of Gearbox Failure

The prediction difference for the GA-PLS outputs trained with SC, takes the form of a fixed beam as shown in Figure 16. One can claim that the gearbox failure can be predicted about 18000 measurements before it happens (around no. 55000), by investigating the hydraulic oil prediction difference and setting an alarm before a higher divergence occurs. In Figure 17, each prediction error value is averaged over the surrounding 144 data points, equal to one day. The smooth green curve has averaged out the peaks, but still shows the tendency of a fixed beam, confirming the idea of being able to predict the gearbox failure.

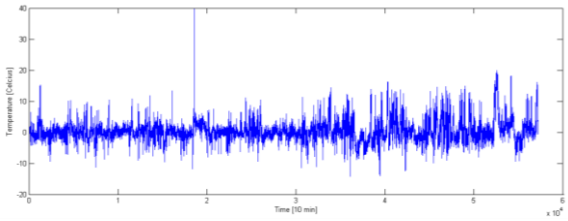

Figure 16: Hydraulic Oil Tº Prediction

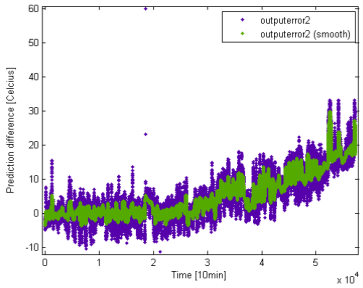

Figure 17: Hydraulic Oil Tº prediction difference smoothened over 1 day

In Figure 18, the hydraulic oil outputs, trained by the SC method are plotted along with the gearbox bearing temperature raw data reduced by halves. For turbine 1, the hydraulic oil prediction difference hits 10 degrees at about 20000 data points (139 days) before the breakdown. For turbine 2, the hydraulic oil prediction difference hits 10 degrees 5000 points (35days) before the gearbox failure. It is worth noting that 10 degrees was here selected as an anomaly threshold and it can be changed by the operator.

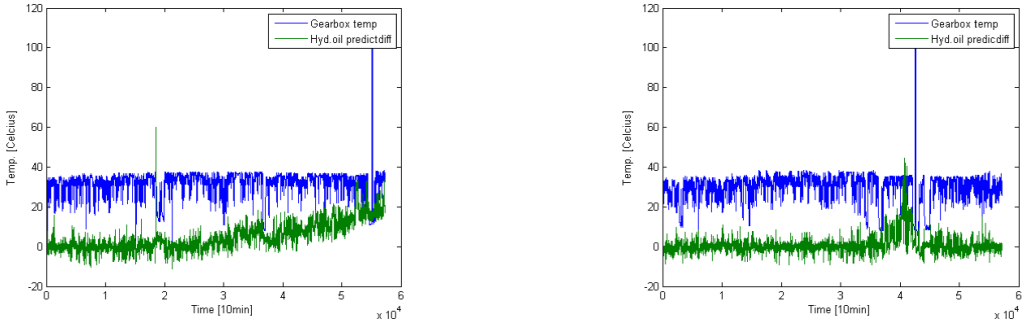

Figure 18: Gearbox raw data /2 and hyd. oil prediction diff. for turbine 1 and turbine 2

Here, the very basic steps of building a condition monitoring system for a CPS (off-shore wind turbine farm) are presented. All the tools used can be easily retrieved from MATLAB tool boxes which make the approach very easy to reproduce. When dealing with CPS, the interconnected objects can yield huge amounts of data to be treated; as it is the case here, it becomes tedious numerically to predict failure using all of the available information. The importance of reducing the dimensionality of the features representing the physical behaviour becomes crucial for speeding and optimizing the learning [ABB+06]. Simpler behavioural

models for condition monitoring can still predict failure, sometimes even better than more complex ones [ABB+03], with enough days to act on the CPS maintenance either automatically by performing a remote action or in last call cases by sending an engineer in case of total failure.

4.4. Maintenance concepts for CPS

From the viewpoint of product life cycles, CPSs can be designed, manufactured, maintained, and treated at the end of life different from conventional products and systems.

Since maintenance is perhaps the most labour intensive, and hence expensive life cycle phase, robotic maintenance is considered promising. The use of robots in maintenance is, however, still limited to inspection, monitoring, or servicing. Many of them are remotely controlled and often dragging power cables [FBK+15, DAR16]. There are various reasons for this.

Maintenance is considered as difficult to automate for four reasons [TKH+04, FaT014]. First, maintenance is irregular in that failures happen stochastically. Second, maintenance is non-uniform because every failure is different. Third, maintenance is non-deterministic, because a maintenance operation may change the system's state and the consequence of one operation determines subsequent maintenance operations. Forth, maintenance is not standardised and requests often specialised tools and methods. Due to these reasons, intelligence for maintenance must deal flexibly with a variety of situations. Therefore, a general purpose maintenance robot needs huge general maintenance intelligence, but it is extremely difficult to develop such general intelligence that can deal with almost any failures and situations.

Consequently, the combination of "*robotic maintenance*" and "*autonomous maintenance*" could be promising [FBK+15]. Autonomous maintenance is a way of maintenance performed by a system that is equipped with sensors, controller, and actuators and responding to the internal and environmental changes without human interventions. This is exactly what CPS can easily achieve – being equipped with maintenance intelligence specialised for the system, rather than the maintenance robot is equipped with versatile maintenance intelligence. In this way the necessary knowledge can be limited only to about the system to be maintained. This local intelligence controls and conducts maintenance by collecting data through embedded sensors and by using robots as a way to deliver maintenance capabilities (such as inspection, exchanging components, adjustments, etc.).

When a CPS detects a fault, it tries to diagnose itself using embedded intelligence and to repair the fault. The repair process will be performed by various methods, including a maintenance robot that delivers physical repair methods. However, the robot does not have specific maintenance knowledge. Instead, the system issues maintenance instructions, such as opening an inspection door, topping up fluid, changing filters, measuring and inspecting the object system. These instructions may include position, specific actions, and motion for the robot arm to operate. In this way, the robot does not have to know "*general maintenance knowledge*".

5. Conclusions

This paper has discussed the importance of Design, Modelling, Simulation and Integration of Cyber Physical Systems. The important aspects which we focus on are methods and applications. Future research directions will focus on trends of evolution, roadmaps of development, social embedding, and/or expected impacts of CPSs. The recommendations presented in this survey are important for researchers working in the context of multi-disciplinary systems. We hope to have demonstrated the different levels of granularity and consideration through the selected case studies. We also hope that these reflections will stimulate the different community involved in the design of Cyber Physical Systems consider moving in this integration direction. As highlights can be pointed out the presentation of a systematic classification of systems, the systematic review of CPS-Design literature with an emphasis on the design, modelling, simulation and integration of CPS and the formulation of architectural and behavioural paradigms for CPS.

Acknowledgments

This work was in part supported by the Linz Center of Mechatronics (LCM) within the framework of the Austrian COMET-K2 program. The authors thank all involved partners for their support.

References

[ABB+03] S. Achiche, M. Balazinski, L. Baron, K. Jemielniak, Tool Wear Monitoring Using Genetically-Generated Fuzzy Knowledge Bases, Engineering Applications of Artificial Intelligence 15 (2003) 303-314.

[ABB+06] S. Achiche, L. Baron, M. Balazinski, M. Benaoudia, Online prediction of Pulp Brightness Using Fuzzy Logic Models, Engineering Applications of Artificial Intelligence 20 (2006) 25-36.

[Aca12] Acatech, Discussion Paper, ProCPS – Production CPS, 2012, http://www.acatech.de/de/publikationen/materialienbaende/uebersicht/detail/artikel/forschungsfragen-in-produktionsautomatisierung-der-zukunft.html

[AcAp10] S. Achiche, F.P. Appio, Fuzzy Decision Support in the Early Phases of the Fuzzy Front End of Innovation in Product Development, in: Proc. of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference: IDETC/CIE 2010. Montreal, Canada, 2010.

[AcTo16] S. Achiche, T. Tomiyama, Design of Multidisciplinary Cyber Physical Systems, Journal of Integrated Design and Process Science, 19 2016, 1-3.

[AlAh14] M.A. Al Faruque, F. Ahourai, A Model-Based Design of Cyber Physical Energy Systems, in: Proc. of 19th Asia and South Pacific Design Automation Conference, SunTec, Singapore, 2014.

[APA13] R. Anderl, A. Picard, K. Albrecht, Smart Engineering for Smart Products, Smart Product Engineering. Abramovici M., Stark R. (ed.), Springer Berlin, Germany, 2013.

[ASR+16] S. Achiche, M. Schlechtingen, M. Raison, L. Baron, I. Santos, Adaptive Neuro-Fuzzy Inference System Models for Force Prediction of a Mechatronic Flexible Structure, Journal of Integrated Design and Process Science 19 (2016) 77-94.

[BFS12] M. Bonfè, C. Fantuzzi, C. Secchi, Design patterns for model-based automation software design and implementation, Control Engineering Practice (2012).

[BHV14] I. Bojanova, G. Hurlburt, J. Voas, Imagineering an Internet of Anything, Computer (2014) 47(6):72-77.

[BLE16] M. Bricogne, J. Le Duigou, B. Eynard, Design processes of mechatronic systems, Mechatronic Futures, Hehenberger P. & Bradley D. (Eds), Springer, London, 2016.

[BTE+10] F. Belkadi, N. Troussier, B. Eynard, E. Bonjour, Collaboration based on Product Lifecycles Interoperability for Extended Enterprise, International Journal on Interactive Design and Manufacturing 4 (2010) 169 -179.

[BTR+14] M. Bricogne, N. Troussier, L. Rivest, B. Eynard, Concurrent versioning principles for collaboration: towards PLM for hardware and software data management, International Journal of Product Lifecycle Management 7 (2014) 17–37.

[CAR14] A. Canedo, M.A. Al Faruque, J.H. Richter, Multi-disciplinary integrated design automation tool for automotive Cyber Physical systems, in: Proc. of Design, Automation and Test in Europe Conference and Exhibition. Dresden, Germany, 2014.

[Car15] N. Carr, The Glass Gage: Where Automation is Taking Us, Bodley Head, 2015.

[CPdA+12] I. Calvo, E. Portillo, O. de Albeniz, A. Armentia, M. Marcos, E. Estèvez, R. Marau, L. Almeida, P. Pedreiras, Towards an Infrastructure Model for Composing and Reconfiguring Cyber Physical Systems, Ubiquitous Computing and Ambient Intelligence, Lecture Notes in Computer Science, J. Bravo, D. Lopez-de Ipina, und F. Moya, Eds., Springer Berlin Heidelberg, 2012, 282–289.

[DAR16] DARPA 2016, http://www.darpa.mil/program/darpa-robotics-challenge, accessed on 18 February 2016.

[DOB+13] G. Di Orio, J. Barata, C. Sousa, L. Flores, Control System Software Design Methodology for Automotive Industry, in: Proc. of IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2013.

[DoCh11] N. Do, G. Chae, A Product Data Management architecture for integrating hardware and software development, Computers in Industry 62 (2011) 854–863.

[EGN+04] B. Eynard, T. Gallet, P. Nowak, L. Roucoules, UML based Specifications of PDM Product Structure and Workflow, Computers in Industry 55 (2004) 301-316.

[EGT+15] S. El Kadiri, B. Grabot, K.D. Thoben, K. Hribernik, Ch. Emmanouilidis, G. von Cieminski, D. Kiritsis, Current trends on ICT technologies for enterprise information systems, Computers in Industry (2015).

[EGV+11] A. Etienne, E. Guyot, D. Van Wijk, L. Roucoules, Specifications and development of interoperability solution dedicated to multiple expertise collaboration in a design framework, International Journal of Product Lifecycle Management 5 (2011) 272–294.

[Eva11] D. Evans, The Internet of Things - How the Next Evolution of the Internet Is Changing Everything, www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, 2011.

[FaTo14] M. Farnsworth, T. Tomiyama, Capturing, classification and concept generation for automated maintenance tasks, CIRP Annals Manu Tech; 63 (2014) 149–152.

[FBK+15] M.J. Farnsworth, C. Bell, S. Khan, T. Tomiyama, Autonomous maintenance for through-life engineering. In: Redding L, Roy R, editors. Through-life engineering services. Switzerland: Springer International Publishing; 2015, 395-419.

[FeZh14] S. Feng, L. Zhang, Model transformation for cyber physical systems, Advances in Computer Science and its Applications, Jeong , H., et al, (ed.), Springer, Berlin, Germany, 2014.

[FHK+15] Feldmann, S.; Herzig, S.; Kernschmidt, K.; Wolfenstetter, T.; Kammerl, D.; Qamar, A.; Lindemann, U.; Krcmar, H.; Paredis, C. and Vogel-Heuser, B.: A Comparison of Inconsistency Management Approaches Using a Mechatronic Manufacturing System Design Case Study. In: 10th IEEE International Conference on Automation Science and Engineering (CASE 2015), Gothenburg, Sweden, 2015, PP. 158-165

[FoVo12] J. Folmer, B. Vogel-Heuser, Computing dependent industrial alarms for alarm flood reduction, in: Proc. of International Multi-Conference on Systems, Signals & Devices 2012, 1–6.

[FoVo13] J. Folmer, B. Vogel-Heuser, Computing Dependent Industrial Alarms for Alarm Flood Reduction. Transactions on Systems, Signals and Devices (2013).

[FPS11] U. Frank, J. Papenfort, and D. Schütz, Real-time capable software agents on IEC 61131 systems – Developing a tool supported method, in: Proc. of World Congress of the International Federation of Automation Control (IFAC) 2011 9164–9169.

[Fri11] P. Fritzson, Modelica - A Cyber Physical Modeling Language and the OpenModelica Environment, in: Proc. of 7th International Wireless Communications and Mobile Computing Conference, Turkey, Istanbul, 2011.

[GAW14] M. Grimm, R. Anderl, Y. Wang, Conceptual approach for multi-disciplinary cyber physical systems design and engineering, in: Proc. of TMCE 2014, Budapest, Hungary, 2014.

[GLA07] N. Gronau, A. Lämmer, K. Andresen, Entwicklung wandlungsfähiger Auftragsabwicklungssysteme, GITO-Verlag, Berlin, 2007.

[Gol89] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, Inc., 1989.

[HeEl11] D. Henriksson, H. Elmqvist, Cyber Physical Systems Modeling and Simulation with Modelica, in: Proc. of 8th International Modelica Conference, Dresden, Germany, 2011.

[Heh12] P. Hehenberger, Advances in model-based mechatronic design, Trauner Verlag, Linz, Austria, 2012.

[HeSo09] F. Heidtmann, D. Soffker, Virtual Sensors for Diagnosis and Prognosis Purposes in the Context of Elastic Mechanical Structures, Sensors Journal IEEE 9 (2009) 1577-1588.

[HTF96] F. Harashima, M. Tomizuka, T. Fukuda, Mechatronics - What Is It, Why, and How? IEEE/ASME Transactions on Mechatronics (1996), 1-4.

[INC10] INCOSE, INCOSE requirements management tools survey, https://www.innoslate.com/incose-requirements-management-tool-survey-response/, 2010.

[Industrie4.0] https://en.m.wikipedia.org/wiki/Industry_4.0 (accessed 27th January 2016)

[ISO4210] ISO/IEC/IEEE 42010, http://www.iso-architecture.org/42010/.

[JoFo13] T.A. Johansen, T. I. Fossen, Control allocation—A survey, Automatica, 49 (2013) 1087–1103.

[Kol14] K. Kołowrocki, Reliability of Large and Complex Systems (Second Edition), Elsevier, Oxford, (2014).

[KäVä09] J. Kääriäinen, A. Välimäki, Applying Application Lifecycle Management for the Development of Complex Systems: Experiences from the Automation

Industry. Software Process Improvement, 42, Berlin, Heidelberg: Springer, 149-160, 2009.

[KKV13] B. Kayyali, D. Knott, S. Van Kuiken, The big-data revolution in US health care: Accelerating value and innovation, McKinsey & Co, (2013), 1-13.

[KoTo12] H. Komoto, T. Tomiyama, A framework for computer-aided conceptual design and its application to system architecting of mechatronics products, Computer-Aided Design, 44 (2012) 931-946.

[KWM+13] Kernschmidt, K.; Wolfenstetter, T.; Münzberg, C.; Goswami, S.; Lindemann, U.; Krcmar, H.; Vogel-Heuser, B.: Concept for an Integration-Framework to enable the crossdisciplinary Development of Product-Service Systems. In: IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, 2013

[LCB+14] J. Lefèvre, S. Charles, M. Bosch-Mauchand, B. Eynard, E. Padiolleau, Multidisciplinary Modelling and Simulation for Mechatronic Design, J. Design Research, 12 (2014) 127-144.

[Lee08] E.A. Lee, Cyber Physical Systems: Design Challenges, In: Proc. 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC), 2008.

[LeGo98] R. Leardi, A.L. González, Genetic algorithms applied to feature selection in pls regression: how and when to use them. Chemometrics and Intelligent Laboratory Systems (1998) 195–207.

[LeRo12] P. Leitão, N. Rodrigues, Modelling and validating the multi-agent system behaviour for a washing machine production line, in: Proc. of IEEE Symposium on Industrial Electronics (ISIE), 2012.

[LeSe11] E. Lee, S. Seshia, Introduction to embedded systems: a Cyber Physical systems approach, Lee & Seshia, USA, 2011.

[Lev12] N.G. Leveson, Engineering a Safer World, MIT Press, 2012.

[LMV13] P. Leitão, V. Marik, P. Vrba, Past, Present, and Future of Industrial Agent Applications. In: IEEE Trans. Ind. Informatics 9 (2013) 2360–2372.

[LSM10] J. Lin, S. Sedigh, A. Miller, Modeling Cyber Physical Systems with semantic agents, in: Proc. of 34th Annual Computer Software and Applications Conference Workshops, Seoul, Korea, 2010.

[LSVH13] C. Legat, D. Schütz, B. Vogel-Heuser, Automatic Generation of Field Control Strategies for Supporting (Re-)Engineering of Manufacturing Systems, Journal of Intelligent Manufacturing, 25 (2014) 1101-1111.

[LZV+11] W. Lepuschitz., A. Zoitl, M. Vallée, M. Merdan, Toward Self-Reconfiguration of Manufacturing Systems Using Automation Agents, IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews 41 (2011) 52–69.

[Mac56] R.H. Macmillan, Automation: Friend or Foe, Cambridge University Press, 1956.

[MVL11] V. Marik, P. Vrba, P. Leitão (Eds.), 5th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS, Vol. 6867 of Lecture Notes in Artificial Intelligence, Toulouse, France, 2011.

[NSE15] S. Nazari, C. Sonntag, S. Engell, A Modelica-based modeling and simulation framework for large-scale Cyber Physical Systems of Systems, IFAC-Papers Online, 48 (2015) 920-921.

[OMG09] Object Management Group, Systems modeling language specification, http://www.omg.org/spec/SysML/1.2/PDF/, 2009.

[PeAu13] L. Petnga, M. Austin, Ontologies of Time and Time-based Reasoning for MBSE of Cyber Physical Systems, in: Proc. of Conference on Systems Engineering Research, Atlanta, USA, 2013.

[PoHo16] S. Pourtalebi, I. Horváth, Towards a methodology of system manifestation features-based pre-embodiment design, Journal of Engineering Design (2016), http://dx.doi.org/10.1080/09544828.2016.1141183

[PSL11] C. Popescu, M. Sotob, J. Lastra, A Petri net-based approach to incremental modelling of flow and resources in service-oriented manufacturing systems, International Journal of Production Research 50 (2011) 325–343.

[RGL+12] N. Raveendranathan, S. Galzarano, V. Loseu, R. Gravina, R. Giannantonio, R., M. Sgroi, R. Jafari, and G. Fortino, From Modeling to Implementation of Virtual Sensors in Body Sensor Networks, Sensors Journal,IEEE 12 (2012) 583-593.

[RSB+08] S. Rachuri, E. Subrahmanian, A. Bouras, S. Fenves, S. Foufou, R. Sriram, Information sharing and exchange in the context of product lifecycle management: role of standards, Computer-Aided Design 40 (2008) 789–800.

[SBD+09] M.M. da Silva, O. Brüls, W. Desmet, H. Van Brussel, Integrated structure and control design for mechatronic systems with configuration-dependent dynamics, Mechatronics 19 (2009) 1016–1025.

[ScSa10] M. Schlechtingen, I. Santos, Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection, Technical report, Department of Mechanical Engineering, Section of Solid Mechanics, Technical University of Denmark, 2010.

[SeCa12] G. Şenaltun, C. Cangelir, Software Management in Product Structure, In: Product Lifecycle Management Toward Knowledge-Rich Enterprise, 388, (2012) 369–378.

[SHY+06] W. Shen, Q. Hao, H.J. Yoon, D.H. Norrie, Applications of agent-based systems in intelligent manufacturing: An updated review, Advanced engineering informatics 20 (4) (2006) 415–431.

[SRF+05] E. Subrahmanian, S. Rachuri, S. Fenves, S. Foufou, R.D. Sriram, Product lifecycle management support: a challenge in supporting product design and manufacturing in a networked economy, International Journal of Product Lifecycle Management, 1 (2005) 4–25.

[SSA13a] M. Schlechtingen, I. Santos, S. Achiche, Wind turbine condition monitoring based on SCADA data using normal behavior models: Part 1: System description, Applied Soft Computing, 13 (2013) 259-270.

[SSA13b] M. Schlechtingen, I. Santos, S. Achiche, Using Data-Mining Approaches for Wind Turbine Power Curve Monitoring: A Comparative Study, IEEE Transactions on Sustainable Energy 4 (2013) 671-679.

[Süt10] M. Süttmann, Master Thesis: Condition Monitoring in Wind Turbines - A Drive Train Monitoring System, 2010.

[SWL+13] D. Schütz, A. Wannagat, C. Legat, B. Vogel-Heuser, Development of PLC-based Software for Increasing the Dependability of Production Automation Systems, IEEE Transactions on Industrial Informatics 9 (2013) 2397-2406.

[SWL+13] D. Schütz, A. Wannagat, C. Legat, B. Vogel-Heuser, Development of PLC-Based Software for Increasing the Dependability of Production Automation Systems, IEEE Transactions on Industrial Informatics 9 (2013) 2397–2406.

[TBD+10] S. Terzi, A. Bouras, D. Dutta, M. Garetti, D. Kiritsis, Product lifecycle management – from its history to its new role, International Journal of Product Lifecycle Management, 4 (2010) 360-389.

[TBK+13] B. Tabbache, M.E.H Benbouzid, A. Kheloui, and J. Bourgeot, Virtual-Sensor-Based Maximum-Likelihood Voting Approach for Fault-Tolerant Control of Electric Vehicle Powertrains, IEEE Transactions on Vehicular Technology 62 (2013) 1075-1083.

[Thr13] K. Thramboulidis, IEC 61499 as an Enabler of Distributed and Intelligent Automation: A State-of-the-Art Review—A Different View, Journal of Engineering (2013) 9.

[TKH+04] S. Takata, F. Kimura, F.J.A.M. van Houten, E. Westkamper, M. Shpitalni, D. Ceglarek, J. Lee, Maintenance: changing role in life cycle management. CIRP Annals Manu Tech 53 (2004) 643-655.

[TVK09] S. Theiss, V. Vasyutynskyy, K. Kabitzsch, Software agents in industry: A customized framework in theory and praxis, IEEE Transactions on Industrial Informatics 5 (2009) 147–156.

[USV14] S. Ulewicz, D. Schütz, B. Vogel-Heuser, Integration of Distributed Hybrid Multi-Agent Systems into an Industrial IT Environment, in: Proc. of 12th IEEE International Conference on Industrial Informatics (INDIN), Porto Alegre, 2014.

[VDI10] VDI/VDE, VDI/VDE: guideline 2653 – multi agent systems in industrial automation – part 1: Fundamentals. 2010.

[VeEh12] J.A. Venter, E.M Ehlers, An architecture description language for modelling dynamic Cyber-Physcal Architecture: a case study, in: Proc. of TMCE 2012. Budapest, Hungary, 2012.

[VGL15] B. Vogel-Heuser, P. Göhner, A. Lüder, Agent based control of production systems - and its architectural challenges, Industrial Agents: Emerging Applications of Software Agents in Industry, Elsevier, 2015.

[VHSL+14] B. Vogel-Heuser, D. Schütz, C. Legat, T. Frank, Model-Driven Engineering of Manufacturing Automation Software Projects - a SysML-based Approach, Mechatronics 24 (2014) 883-897.

[VLL15] B. Vogel-Heuser, J. Lee, P. Leitao, Agents enabling Cyber Physical Production Systems, Automatisierungstechnik (at) 63 (2015) 777-789.

[Vog15] B. Vogel-Heuser, Cyber Physical Production systems / Industry 4.0 - challenges in research and industrial application, 41st Annual Conference of the IEEE Industrial Electronics Society (IECON), Yokohama, Japan, Plenary Talk, 2015, available at https://mediatum.ub.tum.de/node?id=1283333

[VoHe16] B. Vogel-Heuser, D. Hess, Guest Editorial Industry 4.0–Prerequisites and Visions, IEEE Transactions on automation Science and Engineering 13 (2016) 411.

[Vya11] V. Vyatkin, IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review, IEEE Transactions on Industrial Informatics 7 (2011) 768–781.

[Wag03] T. Wagner, An agent-oriented approach to industrial automation systems, in: Agent Technologies, Infrastructures, Tools, and Applications for E-Services, Springer, 2003, 314–328.

[Wan02] L. Wang, Collaborative conceptual design state of the art and future trends, Computer-Aided Design, 34 (2002) 981-996.

[WaVo08] A. Wannagat, B. Vogel-Heuser, Increasing flexibility and availability of manufacturing systems - Dynamic reconfiguration of automation software at runtime on sensor faults, in: Proc. of 9th IFAC Workshop on Intelligent Manufacturing Systems (IMS), Szczecin, 2008, 278-283.

[Wei99] G. Weiss, Multiagent systems: a modern approach to distributed artificial intelligence, The MIT press, 1999

[WiVo11] D. Witsch, B. Vogel-Heuser, PLC-Statecharts: An Approach to Integrate UML-Statecharts in Open-Loop Control Engineering – Aspects on Behavioral Semantics and Model-Checking, in: Proc. of 18th IFAC World Congress (IFAC), 2011, 7866-7872.

[WMM+11] M. Wächter, P. Milan, T. Mücke, J. Peinke, Power performance of wind energy converters characterized as stochastic process: Applications of the Langevin power curve, Wind Energy, 14 (2011) 711–717.

[Wor15] World Economic Forum, Industrial Internet of Things: Unleashing the Potential of Connected Products & Services, www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf, 2015.

[ZBL+14] C. Zheng, M. Bricogne, J. Le Duigou, B. Eynard, Survey on mechatronic engineering: A focus on design methods and product models, Advances in Engineering Informatics 28 (2014) 241–257.

[ZLH+16] C. Zheng, J. Le Duigou, P. Hehenberger, M. Bricogne, B. Eynard, Multidisciplinary integration during the conceptual design process: a survey on design methods of Cyber Physical systems, in: Proc. of DESIGN 2016 14th International Design Conference, May 16-19 2016, Dubrovnik, Croatia.

CVs

Peter Hehenberger, Institute of Mechatronic Design and Production, Johannes Kepler University Linz, Austria

Peter Hehenberger works as Assistant Professor at the Institute of Mechatronic Design and Production (IMDP) at Johannes Kepler University (JKU), Linz, Austria. He received his Diploma degree and his Doctorate degree in Mechatronics from JKU in 2000 and 2004, respectively. His core competencies cover “model-based mechatronic system design” and he has published over 70 peer-reviewed papers and been guest editor for special issues. Among other scientific service activities Peter Hehenberger organizes the “International Workshop Series on Mechatronic Design” (Linz 2012, Paris 2013, Leuven 2014). He is also member of VDI, ASME, IFIP WG5.1 (“Global Product development for the whole life-cycle”) and Design Society, where he co-chairs a Special Interest Group (SIG) on Methodologies for Design, Integration, Modelling and Simulation of Cyber Physical Systems.

Birgit Vogel-Heuser, Institute of Automation and Information Systems, Technische Universität München, Garching, Germany

Birgit Vogel-Heuser is a Full Professor and Director with the Institute of Automation and Information Systems, Technical University of Munich, Germany.

She graduated in electrical engineering, holds a PhD in mechanical engineering and worked for nearly 10 years in the industrial automation for machine and plant manufacturing industry. After holding different chairs of automation in Hagen, Wuppertal, and Kassel, Germany, she has been the Head of the Automation and Information Systems Institute, Technical University of Munich since 2009. Her research interests include modeling and education in automation engineering for hybrid process and heterogeneous distributed and intelligent systems using a human centered approach.

She is speaker of the Collaborative Research Area CRC 768 on “Managing Cycles in Innovation Processes – Integrated Development of Product-Service Systems Based on Technical Products", member of the coordination board of the SPP 1593 managed software evolution, design for future and member of acatech.

David Bradley, Abertay University, Dundee, UK

David Bradley graduated from the University of Bradford in 1968 having studied electrical engineering as a student apprentice with the Yorkshire Electricity Board. He then remained at Bradford to complete a PhD working on the modelling of current transformer behaviour in relation to power system protection.

In 1991, with Professors Jim Hewitt and Jack Dinsdale, David was instrumental in establishing the UK Mechatronics Forum, now the Mechatronics Forum. He was appointed Emeritus Professor at Abertay University in September 2014. Professor Bradley is the author, co-author or editor of seven books, including two on the design of mechatronic systems, and has been involved in 45 journal papers and 120 refereed conference papers and book chapters.

Benoit Eynard, Sorbonne Universites, Universite de Technologie de Compiegne, Department of Mechanical Systems Engineering, France

Benoît Eynard is currently the Director of the Institute for Mechatronics at the Université de Technologie de Compiègne – UTC. In 2007, he has joined UTC as a Full Professor for leading the Department of Mechanical Systems Engineering until 2012. He is also member of the UMR 7337 Roberval UTC/CNRS. Previously, he worked as Assistant Professor at the Université de Technologie de Troyes where he managed the MSc degree in Information Technology for Mechanical Engineering. Since 2013, he is the Director of the AIP-PRIMECA network –French Academic Group on Factories of the Future: integrated design and advanced manufacturing. He is also a member of the Design Society and of IFIP5.1. He is a recognized researcher in product lifecycle management and data exchange, collaborative engineering and digital manufacture, eco-design and sustainable manufacturing.

Tetsuo Tomiyama, Cranfield University, Bedfordshire, UK

Tetsuo Tomiyama has been Professor of Life Cycle Engineering at Cranfield University since October 2012. Prior to this appointment, he was Professor at Delft University of Technology in The Netherlands (between 2002 and 2012), Professor (between 1998 and 2002) and Associate Professor (between 1987 and 1998) at the University of Tokyo in Japan. He holds BSc, MSc, and PhD (all from the University of Tokyo). He is currently a fellow of CIRP, ASME, and JSME, as well as a member of internationally recognised organisations including IEEE, AAAI and the Design Society.

Professor Tomiyama has co-authored a number of research papers, edited books, and been invited to a number of prestigious international conferences. He is Emeritus Editor of the International Journal on Advanced Engineering Informatics. He is also serving as editorial member for a number of other renowned international scientific journals. In May 2012, he received a Royal Society Wolfson Research Merit Award.

Sofiane Achiche, Ecole Polytechnique de Monteral, Quebec, Canada

Sofiane Achiche works as a Professor at the Department of Mechanical Engineering in École Polytechnique de Montréal, Canada. He conducted his PhD in Mechanical Engineering from École Polytechnique de Montréal. He also works in the field of evolutionary computational intelligence for control and decision support applied to engineering problems such as: emotional design, process control, measurements and mechatronic systems. His research interests focus upon understanding and modelling activities of new product development processes for decision support purposes.

https://dspace.org/

Design, modelling, simulation and integration of cyber physical systems: Methods and applications

Hehenberger, P.; Vogel-Heuser, B.; Bradley, D.; Eynard, B.; Tomiyama, Tetsuo ...

2016-07-18
Elsevier
http://dx.doi.org/10.1016/j.compind.2016.05.006
Hehenberger P, Vogel-Heuser B et al. Design, modelling, simulation and integration of cyber physical systems: Methods and applications, Computers in Industry, Volume 82, October 2016, Pages 273–289.

Downloaded from a DSpace repository. https://dspace.org/